虚拟仪器驱动器综述

刘金宁[1]，孟 晨[1]，崔少辉[1]，陈德祥[2]
(1.军械工程学院，河北 石家庄 050003；2.南京军区导弹站，江苏 南京 210028)

摘 要：驱动器是连接虚拟仪器物理硬件和测试应用程序的桥梁和纽带，回顾了虚拟仪器驱动器技术规范的发展历程。在分析规范的基础上给出了两种驱动器设计类型，预测了其今后的发展趋势。分析了驱动器发展过程中的相关支撑技术，论述了我国的研究现状，指出基于组件技术的信号型驱动器设计是虚拟仪器驱动器的发展方向。

关键词：虚拟仪器 驱动器 信号接口 COM 软件工程

计算机在测试和自动化领域中的应用，导致了仪器“驱动器”概念的诞生，驱动器又称驱动程序。仪器驱动器是介于计算机与仪器硬件设备之间的软件中间层，由函数库、实用程序、工具套件等组成，是一系列软件代码模块的统称。它驻留在计算机中，是连接计算机和仪器的桥梁和纽带。采用驱动器可以使计算机有能力控制物理仪器设备，随着 VXI、PXI 等标准总线的出现，开创了测试系统发展的崭新空间——虚拟仪器（Virtual Instruments）。虚拟仪器代表着从传统硬件为主的测试系统到以软件为中心的测试系统的根本性转变。

1 技术规范回顾

计算机在测试领域的应用经历了总线型仪器、PC 仪器、虚拟仪器等不同的发展阶段。伴随着这一过程，仪器驱动器技术规范以通用性为基本出发点，仪器互换性和互操作性以及软件移植性为根本指导原则，从最初的 IEEE-488.2、SCPI (Standard Command for Programming Instrument)发展到现在的 IVI-MSS（Measurement and Stimulus Subsystem）、IVI-Signal Interface，已经走过了艰辛而漫长的历程。它们建立在 Windows 操作系统驱动程序设计模式 VxD 和 WDM（Windows Driver Model）之上，并融入了仪器操作的具体内容。

1.1 IEEE 488.2

IEEE-488 是 1975 年由 IEEE 发布的一个重要的仪器控制总线标准。IEEE-488.1 定义了计算机和仪器之间的硬件接口规范；IEEE-488.2 定义了 TPS（Test Program Set）和仪器之间的软件接口规范。IEEE-488.2 规定了数据代码和格式，用一组公用命令和协议定义了测试系统中控制器和仪器之间的通信标准，共有 39 条，这些命令提供了仪器的内部管理功能。IEEE-488.2 没有严格的语义定义，同样的功能不同厂商可用不同的命令来实现，而且这一标准仅适合于 GPIB 类仪器，通用性、互换性很差。

1.2 SCPI

IEEE-488.2 没有涉及为了提供测量和激励所必需的命令。1990 年，在 IEEE-4888.2 标准和 IEEE-754 标准之上，制定了 SCPI 标准。它通过指定一组通用控制命令来实现对多类仪器的相同控制。在仪器功能严格匹配（如具有相同的精确度、测量范围等）的前提下，可实现互换，扩展了仪器互换的空间。然而，这种互换性限制了仪器生产厂家对仪器功能的扩展，实用性差，加上 SCPI 编程的复杂性，通用性、互换性水平较低。

1.3 VPP

1993 年，VPP（VXI Plug & Play）系统联盟发布了 VPP 规范，该规范定义了系统的框架、软件接口、软件环境和仪器驱动器模型。它把与仪器的底层通信封装成一些高层函数，执行仪器的控制功能。VISA（Virtual Instrument Software Architecture）作为底层 I/O 库，是这一时期的主要成果。它不区分仪器的种类，用一组通用函数实现驱动器功能，通用性得到了很大加强。然而，跟 IEEE-488.2 类似，VPP 驱动器接口仍没有严格的语义标准，仪器厂商可以根据自己的特长进行开发，这使得驱动器产品的接口不统一，仪器互换性仍没有最终实现。

1.4 IVI

为了实现仪器互换和互操作，1998 年成立了 IVI（Interchangeable Virtual Instruments）基金会，讨论开发可互换仪器驱动模型，旨在对硬件互换、运行性能、发展弹性、质量保证等驱动器问题进行规范。

IVI 模型是 IVI 基金会在 VPP 技术规范基础上制定的一种驱动器设计标准。它通过定义类驱动器和专用驱动器（独立的软件层）并增加仪器仿真、状态缓存、量程监视等机制实现了部分通用仪器之间的互换，提高了测试程序的开发效率。

然而，面向仪器互换的虚拟仪器设计目标，IVI 模型仍然存在以下不足：

(1)只适合同类仪器的互换，不能实现不同类仪器或某些具备两类、多类仪器功能的综合性仪器之间的互换。

(2)IVI 类驱动器只能统一某类仪器中 80%的仪器功

能，而其它20%功能只能通过专用驱动器来实现。

(3)可用标准较少。目前只完成了示波器、万用表、函数发生器、多路开关等九种仪器的类驱动器的标准化。

(4)标准开放程度低。IVI模型只适合于通用仪器，比如万用表等，而对某些专用仪器(如数据采集卡)不适用。

1.5 IVI-MSS

为了改进IVI模型存在的不足，IVI基金会开始制定IVI-MSS和IVI-Signal Interface规范，它们是在IVI模型的基础上发展起来的，分别实现基于功能和信号的仪器互换操作。其中IVI-MSS于2001年2月发布，现已经是成熟的规范，而IVI-Signal Interface尚待发布。

如图1所示，基于IVI-MSS规范的虚拟仪器测试软件共包括五部分。用户应用程序是IVI-MSS Solution的运行环境，它通过调用IVI-MSS Server提供的编程接口实现对仪器资源的访问；IVI-MSS Server是独立于测试仪器资源的软件层，它封装了测试算法，对外提供面向测试功能需求的编程接口，该接口在被用户应用程序调用时作为“角色”向用户提供测试服务；RCM是连接IVI-MSS Server和仪器Driver的软件层，在RCM内部封装了仪器访问细节，对外提供RCM接口与IVI-MSS Server交互。RCM通过SCPI命令、VISA函数和IVI驱动器等实现对物理仪器的访问。

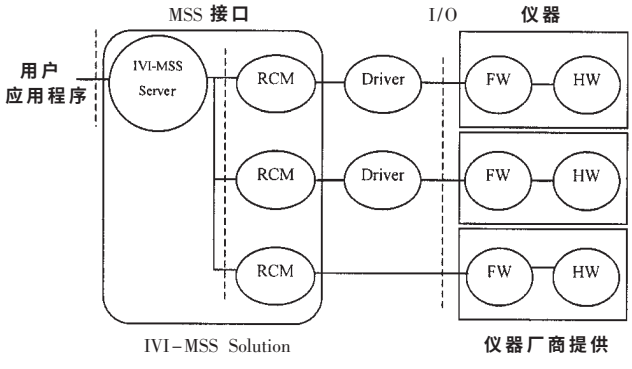

图1 基于IVI-MSS模型的虚拟仪器软件结构

IVI-MSS中IVI-MSS Solution作为一个独立的软件层，为仪器互换提供了解决方案；RCM由开发人员根据需求来定义，对于不同的测试需求，即使是同一测试仪器平台，RCM也是不同的。在更换仪器后，只要提供与原始仪器功能相同或相当的RCM，就可实现相同的测试功能，这样大大拓展了仪器互换的空间。

1.6 IVI-Signal Interface

2000年，IVI基金会的Signal Interface工作组在TYX公司和HP公司的领导下开始制定IVI-Signal Interface标准。它基于COM(Component Object Model)技术，是一系列COM组件的统称。

在IVI-MSS模型基础上发展起来的IVI-Signal Interface标准把原先的仪器控制命令转化为对测试信号的需求，把IVI-MSS Server功能接口进一部封装形成IVI-Signal Interface信号接口。这克服了“面向仪器”的TPS开发中存在的弊端，实现了更高层次的仪器互换。信号接口的标准化增强了不同厂商仪器之间的互操作性，方便了代码移植。同时，为仪器驱动器开发带了巨大商机，提高了IVI信号组件的开发效率和质量，有很看好的应用前景。

综上，当前占主导地位的驱动器设计规范主要有两种：VPP规范和IVI系列规范。两种驱动器开发规范的共同点是均建立在IEEE-488.2和SCPI命令以及VISA库之上，都包括接口和内部实现两部分。不同点是前者已发展成熟，它以仪器本身的特征应用为中心，是功能驱动的，多由仪器生产商提供，接口没有严格的语义标准，实现了仪器的即插即用，没有实现仪器互换和软件移植等功能；而后者建立在VPP之上，正处在发展完善之中，它面向UUT的测试需求，是需求驱动的，由测试系统集成人员或第三方软件开发人员编写，接口有严格的语义标准，部分地实现了仪器互换性和软件移植性，并最终向着完全实现而努力。目前VPP规范已被多数厂家所采纳进行自己产品的驱动器开发，因此将在未来很长的一段时间内占统治地位。但由于它在解决仪器互换性问题上无能为力，随着IVI系列规范的进一步完善，必将被其替代。

2 驱动器开发

根据以上驱动器设计规范发展回顾及分析可知，驱动器开发也分为两种类型：基于VPP规范的即插即用型驱动器开发和基于IVI系列规范的互换型驱动器开发。

2.1 即插即用型驱动器开发

开发基于VPP规范的即插即用型驱动器的过程分为两步。第一是仪器驱动器外部接口的设计，它表示仪器驱动器如何与外部软件接口，通常提供两种方式的接口：程序式开发接口和图形软面板。软件开发者通过程序式开发接口可以理解每个仪器驱动器函数的功能以及在应用程序中如何调用每个函数、另一种接口方式是图形化软面板，通过这一软面板可以直接操作控制物理仪器。第二是要完成仪器驱动器的内部模块设计，实现仪器的硬件功能。使用程序式开发接口的用户了解了这一功能，可以在应用程序中直接应用这些模块，而不必通过软面板操作。

要完成第一项工作应选用界面编辑功能较强的编程环境，减少仪器软面板的开发时间；第二项工作通过调用VISA I/O库中的函数来完成，编程语言按照VPP规程可以选用ANSI C、BASIC或者Ada等。选用合适的图形软件工具可以把这两部分工作集成到一个环境下完成，省去两部分的连接工作，例如NI公司的可视化软件平台LabWindows/CVI。CVI开发环境中两部分组成：用户界面设计器和源代码编辑器。

2.2 互换型驱动器开发

与即插即用型驱动器类似，基于IVI系列规范的互

换型驱动器开发也包括两部分。第一是分析测试系统的功能需求,以功能或信号的形式分类定义驱动器组件的接口。这些接口是对 UUT 测试需求的描述,有严格的语义标准,将不随仪器种类和软件类型而改变,是标准的,通用的。第二是驱动器组件接口的内部实现,它被封装在组件内部,可以根据具体的开发工具和编程人员特长来开发,是非标准的、特殊的。

对于基于 IVI 系列规范的驱动器开发,目前还没有专业的、IVI 基金会指定的开发工具。但由于其采用 COM 技术,因此可以使用任何支持组件开发的编程平台进行开发。设计人员在理解技术规范的基础上可以利用现有的图形化编程工具(比如 VC++、VB 等)进行设计。

需要说明的是,IVI 是介于 VPP 和 IVI-MSS 之间的一个过渡性规范,它既有 IVI-C 的 C 语言形式,也有基于 COM 的 IVI-COM 组件形式,并且接口的严格语义标准目前只发布了八类仪器的技术规范,因此可以根据具体情况选用相应工具进行开发。

3 发展趋势

(1)信号型驱动器

由前面对 IVI-Signal Interface 标准的介绍可知,信号型驱动面向 UUT 的测试需求,是需求驱动的,符合当前计算机体系结构发展趋势,而且实现了更高层次的仪器互换和互操作,通用性好。随着面向信号的商业化虚拟仪器软件开发平台的不断涌现,如 PAWS、ATLAS 2K 等,迫切需要给出标准化的软件开发平台与底层硬件模块之间的接口,而信号型驱动器恰好实现了面向信号的 TPS 开发平台与底层硬件模块的完美对接。基于以上分析,信号型驱动器将是虚拟仪器驱动器设计标准发展的必然结果。

(2)网络化驱动器

网络的普及给各个行业都带来了巨大冲击,测试领域也不例外,网络化虚拟仪器和仪器网络化现已成为当前测试技术的一个研究热点。而要想实现远程控制仪器就必须提供仪器设备的网络化驱动器或在现有仪器驱动器的基础上添加网络化功能。基于此,VXI 联盟(VXI Consortium)提出了 VXI-11 规范,这个规范主要是对 IEEE-488 总线的扩展,也就是通过 TCP/IP 发送 IEEE-488 总线命令。该扩展的目的在于实现远端的客户端通过网络与现场仪器通信并完成测试任务,并且对用户来讲好像是在本地使用仪器一样。随着虚拟仪器技术和网络技术的发展,网络化驱动器将取得更大的发展空间。

(3)VISA 兼容更多的接口类型

计算机接口技术不断发展,涌现了许多商业化 PC 总线,如 USB、Ethernet 等,这也是影响虚拟仪器发展的关键技术之一。由于这些总线最初是为网络 PC 和连接 PC 外设而设计的,要更方便地控制仪器,这些总线需要软件构架来简化通信并与其他一起控制标准兼容。因此,需要对虚拟仪器驱动器标准框架 VISA 进行扩展。

(4)与商业 TPS 开发平台"即插即用"

驱动器是连接计算机和物理仪器的中间环节,是虚拟仪器开发的重要资源。当前的 TPS 开发平台向着集成、高效、商家垄断方向发展,如 HP 公司的 VEE、NI 公司的 LabVIEW、LabWindows/CVI 等。为了扩展自己的市场空间和便于用户开发使用,虚拟仪器硬件设计厂商或第三方软件开发单位应该提供与这些商业开发平台"即插即用"的驱动器。同时,为了加强合作和简化集成,商业 TPS 开发商也应该给出与自己平台兼容的驱动器设计标准并提供相关技术支持。

(5)源代码级开放式结构

当前,从源代码开放的角度来讲,虚拟仪器设计领域存在三种驱动器类型:封闭黑盒型,封装型和开放源代码型。其中封闭黑盒型不提供对源代码的访问,不具备扩展和编写仪器新功能的能力;封装型驱动可作为驱动器二次开发的原始接口或封装器;而开放源代码型驱动原生于相应的开发环境,提供对源代码的完全访问权限,经过优化和改进后能易于使用并具有集成的灵活性,能让开发人员定制自己的功能需求,把开发触角伸向了仪器设备的心脏地带。

虚拟仪器的一个重要特点是硬件的功能由软件来定制,而从某种层面上讲,驱动器是仪器功能的描述和表达。当前多数仪器硬件模块开发厂商在发布自己的产品时,都把驱动器作为"黑盒子"来发布,这不利于客户定制自己需要的功能和进行二次开发。另一方面,由于软件升级换代相当迅速,这也给驱动器开发提出了新的要求。现在多数仪器驱动在向.NET 平台移植时都存在各种各样的困难,而驱动器源代码级开放是解决这些问题的前提。不仅如此,开放源代码型驱动还能简化与仪器硬件的连接,使开发人员不仅是驱动器的使用者而且是拥有者。因此,仪器驱动器采用开放源代码式结构将是一个重要的发展方向。

(6)可视化配置操作

可视化与完备的人机交互能力是现代软件开发的基本要求,作为虚拟仪器核心的驱动器在这方面应能满足客户的更高的需求。把软件开发人员从繁重的代码编写任务中解脱出来,而把主要精力放在测试功能的实现上是驱动器设计迫切需要解决的问题。将仪器连接配置、编写测试代码、测试任务的组合设定等繁琐工作转变成友好人机界面下的鼠标操作,必将简化虚拟仪器系统的集成开发。为此,在驱动器设计领域出现了虚拟资源、虚拟通道等概念。它们把物理资源和通道的信息(比如量程、端口配置等)进行封装,通过友好的界面与开发人员进行信息交互,实现了仪器控制的可视化配置操作。

(7)拓展应用空间

仪器驱动器是计算机控制物理仪器设备的中间环

节，随着虚拟仪器的不断发展，这一思想也拓展了崭新的发展空间。驱动器越来越多地以“服务”的形式为测试程序提供功能调用。如 IEEE 1451.4 标准给模拟传感器定义了电子数据表格并内嵌于其中，测试系统或 TPS 开发平台可以通过模拟传感器提供的数字接口读取电子数据表中的数据并对其进行配置，这在一定意义上也是驱动器应用的新领域。使用基于 IEEE1451.4 设计的智能传感器可以简化传感器的连接过程，实现传感器的“即插即用”和自动配置。ISP（Programmable in-system）技术在电子设计自动化领域得到了广泛应用，基于此，NI 公司提出的 RIO（Reconfiguration I/O）技术完成了物理仪器的在线可编程控制，实现了用户自定义硬件，是仪器驱动器概念的升华。

4 关键技术

（1）COM 技术

COM 技术的核心是组件，组件是可以明确辨识和管理、可以提供某项服务的自包含的软件模块。它封装了一定的数据（属性）和方法（函数），并提供特定接口。开发人员通过访问这些特定接口来使用组件，与其它程序模块通信、交互，实现预期功能。组件是实现仪器驱动器语言、平台无关和网络位置透明的关键技术。

基于组件技术的驱动器模块通过标准接口与其它软件模块通信，各个组件就像挂在“软总线”上一样通过公共通道传递信息。基于此，编程人员可以象“搭积木”似的开发自己的测试程序。更换仪器后，只要驱动器接口不变就不用更改测试程序。使用驱动器组件使得仪器模块的互换性、测试软件的开放性和可重用性得到了根本保证，同时实现了软件开发和应用的不断“迭代和增量”过程。

（2）多线程技术

同步、触发、时序操作是仪器控制的客观要求，而多线程技术是满足这一要求的关键技术。Windows 操作系统是一个多任务、多线程操作系统，实行的是抢先式、多任务工作模式。在 Windows 环境中，每一个测试项目可以由一个线程来代表，这意味着一个测试程序可以同时完成多个测试任务。在多线程执行中系统会根据线程的优先级和同步要求分配时间单元用于执行多个线程，这样实现了多任务分时占有 CPU，可在一个段时间内并行完成多个测试任务。多任务、多线程之间通过同步、通信（如共享内存映射文件、访问共享数据以及使用同一消息队列等）以实现复杂的测试、控制逻辑。

（3）引擎技术

测试程序的仪器操作过程是 TPS 利用驱动器控制硬件仪器的过程。为了优化这一控制过程，需要引擎技术，把软件代码的控制需求转变成实际的物理仪器操作。测试中用到的最多的同步、触发功能，若有多个同步步骤需要连续、高速触发，在这样的情况下，仅需要测试代码去控制是很难满足需求的。为此，可以设计基于引擎技术的同步触发引擎，把测试需求编程一定的序列输入到相应的同步触发引擎中。依据测试程序的执行自行触发这一序列，将大大提高测试效率，满足更高的测试速率要求，使测试程序具有自主触发和时钟路由能力。另外，随着便携式、模块化、嵌入式实时环境对虚拟仪器的要求越来越迫切，还需要开发驱动器在这些不同环境下的运行时引擎，以满足各种需求。综上，引擎技术在测试领域中有很大的发展空间，应倍加重视。

（4）软件工程技术

仪器驱动器是对物理仪器的功能描述，软件工程技术将能保证驱动器设计的功能完备性。1997 年由 OMG（Object Management Group）发布的统一建模语言（UML，unified modeling language）和统一软件开发过程是软件工程领域中的重要成果，标志着面向对象技术走向第二代。UML 支持从系统需求分析到详细设计再到系统的验证测试的全部过程，当出现问题时提供跟踪机制。使用 UML 会帮助设计人员在建造驱动器框架中理解模型，把握仪器的全貌和功能、部件之间的联系，防止过早地陷入各个模块细节中去，有利于提高驱动器软件的质量，缩短研制周期，降低开发费用。统一软件开发过程是用户驱动、以架构为中心、不断迭代和增量过程。基于这一过程，可以设计出功能完备、接口标准、易于升级换代的驱动器。

5 国内现状

我国在虚拟仪器驱动器研究方面取得了一定的进展：成都电子科技大学开发出了具有自主知识产权的 VISA 库；哈尔滨工业大学电气工程系开发的虚拟仪器软件开发平台—ATS95 可以实现对 VXI、GPIB 等总线接口的控制；成都 611 所在引进 PAWS 平台的同时也对面向信号的驱动器设计和平台开发做了一定研究等等。但由于我国介入虚拟仪器研究比较晚，在硬件模块方面没有自己上规模、成系列的产品，导致了测试软件没有全面发展，很多关键技术仍处于起步阶段，在驱动器设计方面没有自己知识产权的技术规范和相关产品，仍需要很长的路要走。鉴于此，我们应在以下方面进行努力：（1）开发自己的总线控制器，占领虚拟仪器技术的心脏地带；（2）设计各种仪器模块产品并形成系列化，降低虚拟仪器系统的集成成本；（3）设计完备成熟的 VISA 库，把握自己的知识产权；（4）开展面向信号的驱动器技术研究，与国际接轨，深入研究虚拟仪器核心技术。

参考文献

1 IVI Foundation. IVI-3 系列规范[EB/OL]. http://www.ivi-foundation. org.

2 National Instrument Corporation. Instrumentation Newsletter. Second Quarter 2004:8~9

3 National Instrument Corporation. Instrumentation Newsletter. Four Quarter 2004:19

4 National Instrument Corporation. Instrumentation Newsletter. Third Quarter 2004:6~7

（收稿日期：2005-02-28）